

Name, Computing-id: Aishwarya Gavili, ag5yy
CS 4740

PART 1:

1. [POSITIVE TEST CASE]

Test picture:







System processing image:

```
[ubuntu@ip-172-31-71-149:~]$ python upload.py
[ubuntu@ip-172-31-71-149:~]$ python test.py
(u'0c22494c-b1db-490e-afb3-aec6d753c2b7', 99.99870300292969)
Jim Ryan
(u'de542978-f8f6-4201-a9ac-5e5455aca411', 99.99909973144531)
Jim Ryan
(u'98ff911f-9107-4894-b37e-cb2ad173423a', 99.99700164794922)
Jim Ryan
[ubuntu@ip-172-31-71-149:~]$
```

Name, Computing-id: Aishwarya Gavili, ag5yy
CS 4740

2. [NEGATIVE TEST CASE]

Test pictures:

| | |
|---|------------------------|
|  | <i>UMD President</i> |
|  | <i>Dean Groves 1</i> |
|  | <i>Dean Groves 2</i> |
|  | <i>Teresa Sullivan</i> |

Name, Computing-id: Aishwarya Gavili, ag5yy
CS 4740



Group picture with random university presidents (Jim Ryan and Timothy Sands not included):

System processing image:

```
[ubuntu@ip-172-31-71-149:~]$ python3 test.py  
ubuntu@ip-172-31-71-149:~$
```

Name, Computing-id: Aishwarya Gavili, ag5yy
CS 4740

3.

The screenshot shows the AWS DynamoDB console interface. The left sidebar contains navigation links for DynamoDB, Tables, Backups, Reserved capacity, Preferences, DAX, Dashboard, Clusters, Subnet groups, Parameter groups, and Events. The main content area displays the 'family_collection' table. At the top, there are two notification banners: 'The preview of the new DynamoDB console is now available' and 'Kinesis Data Streams for DynamoDB is now available'. Below these, there are buttons for 'Create table' and 'Delete table'. A search bar allows filtering by table name, and a dropdown menu shows 'family_collection' selected. The table's 'Items' tab is active, showing a list of 6 items. Each item has a checkbox, a 'RekognitionId' (a long alphanumeric string), and a 'FullName' (either 'Jim Ryan' or 'Timothy Sands'). The bottom of the console shows a footer with 'Feedback', 'English (US)', and copyright information.

| RekognitionId | FullName |
|--------------------------------------|---------------|
| 0c22494c-b1db-490e-afb3-aec8d753c2b7 | Jim Ryan |
| 1a08f472-d66b-4fa3-a789-2f71d2dd6df3 | Timothy Sands |
| 98ff911f-9107-4894-b37e-cb2ad173423a | Jim Ryan |
| c6f91a3f-483d-4842-9ff4-715d6d3f182a | Timothy Sands |
| de542978-f8f6-4201-a9ac-5e5455aca411 | Jim Ryan |
| f4eb409a-a6d3-441f-a33b-fa014a87dada | Timothy Sands |

PART 2:

1.

```
In [1]: bucket = 'ag5yy-sagemaker'
        prefix = 'sagemaker/xgboost_credit_risk'

        # Define IAM role
        import boto3
        import re
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import os
        import sagemaker
        from sagemaker import get_execution_role
        from sagemaker.inputs import TrainingInput
        from sagemaker.serializers import CSVSerializer

        role = get_execution_role()
```

```
In [2]: !wget https://archive.ics.uci.edu/ml/machine-learning-databases/00350/default%20of%20credit%20card%20clients.xls

--2021-04-21 14:10:04-- https://archive.ics.uci.edu/ml/machine-learning-databases/00350/default%20of%20credit%20card%20clients.xls
Resolving archive.ics.uci.edu (archive.ics.uci.edu)... 128.195.10.252
Connecting to archive.ics.uci.edu (archive.ics.uci.edu)|128.195.10.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5539328 (5.3M) [application/x-httpd-php]
Saving to: 'default of credit card clients.xls.2'

default of credit c 100%[=====] 5.28M 7.21MB/s in 0.7s

2021-04-21 14:10:05 (7.21 MB/s) - 'default of credit card clients.xls.2' saved [5539328/5539328]
```

```
In [3]: dataset = pd.read_excel('default of credit card clients.xls')
        pd.set_option('display.max_rows', 8)
        pd.set_option('display.max_columns', 15)
        dataset
```

```
Out[3]:
```

| | Unnamed: 0 | X1 | X2 | X3 | X4 | X5 | X6 | ... | X18 | X19 | X20 | X21 | X22 | X23 | Y |
|-------|------------|-----------|-----|-----------|----------|-----|-------|-----|----------|----------|----------|----------|----------|----------|----------------------------|
| 0 | ID | LIMIT_BAL | SEX | EDUCATION | MARRIAGE | AGE | PAY_0 | ... | PAY_AMT1 | PAY_AMT2 | PAY_AMT3 | PAY_AMT4 | PAY_AMT5 | PAY_AMT6 | default payment next month |
| 1 | 1 | 20000 | 2 | 2 | 1 | 24 | 2 | ... | 0 | 689 | 0 | 0 | 0 | 0 | 1 |
| 2 | 2 | 120000 | 2 | 2 | 2 | 26 | -1 | ... | 0 | 1000 | 1000 | 1000 | 0 | 2000 | 1 |
| 3 | 3 | 90000 | 2 | 2 | 2 | 34 | 0 | ... | 1518 | 1500 | 1000 | 1000 | 1000 | 5000 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 29997 | 29997 | 150000 | 1 | 3 | 2 | 43 | -1 | ... | 1837 | 3526 | 8998 | 129 | 0 | 0 | 0 |
| 29998 | 29998 | 30000 | 1 | 2 | 2 | 37 | 4 | ... | 0 | 0 | 22000 | 4200 | 2000 | 3100 | 1 |
| 29999 | 29999 | 80000 | 1 | 3 | 1 | 41 | 1 | ... | 85900 | 3409 | 1178 | 1926 | 52964 | 1804 | 1 |
| 30000 | 30000 | 50000 | 1 | 2 | 1 | 46 | 0 | ... | 2078 | 1800 | 1430 | 1000 | 1000 | 1000 | 1 |

30001 rows x 25 columns

2.

```
In [13]: cm = pd.crosstab(index=test_data['Y'], columns=np.round(predictions), rownames=['Observed'], colnames=['Predicted'])
tn = cm.iloc[0,0]; fn = cm.iloc[1,0]; tp = cm.iloc[1,1]; fp = cm.iloc[0,1]; p = (tp+tn)/(tp+tn+fp+fn)*100
print("\n{0:<20}{1:<4.1f}%\n".format("Overall Classification Rate: ", p))
print("{0:<15}{1:<15}{2:>8}".format("Predicted", "No Default", "Default"))
print("Observed")
print("{0:<15}{1:<2.0f}% ({2:<}){3:>6.0f}% ({4:<})".format("No Default", tn/(tn+fn)*100,tn, fp/(tp+fp)*100, fp))
print("{0:<16}{1:<1.0f}% ({2:<}){3:>7.0f}% ({4:<}) \n".format("Default", fn/(tn+fn)*100,fn, tp/(tp+fp)*100, tp))
```

Overall Classification Rate: 82.1%

| Predicted | No Default | Default |
|------------|------------|-----------|
| Observed | | |
| No Default | 84% (2245) | 33% (107) |
| Default | 16% (429) | 67% (220) |

3. *What percent of the time did you predict a person would default on their credit card payment and they actually did default?*

67%

4. *What percent of the time did you predict a person would NOT default on their credit card and they actually did NOT default?*

84%